Integrated Packet Level FEC Coding for Bulk Data Transfer over Satellite Networks

Gun Akkor[†], John S. Baras[‡], and Michael Hadjitheodosiou[•]
Electrical and Computer Engineering Department and Institute for Systems Research, University of Maryland at College Park, College Park, MD 20742, USA.

Packet level forward error correction (FEC) coding has been widely accepted as an efficient way of reducing the number of retransmissions and recovering from packet losses for reliable multicast in satellite networks. While there has been a significant effort in defining packet level FEC schemes that detail the structure, organization and delivery of the encoding packets, little attention has been given to how the physical channel, channel coding, and the downlink power budget affect the design space of these schemes in the context of a satellite network architecture. In this paper, we take a step in this direction, and define the points of interaction between various components and build a simulation environment. Using this simulation setup, we provide design suggestions for integrating FEC coding into reliable multicast transport protocols.

Nomenclature

FEC	=	Forward Error Correction
DVB	=	Digital Video Broadcasting
TCP	=	Transmission Control Protocol
IP	=	Internet Protocol
MPEG	=	Moving Picture Experts Group
EIRP	=	Equivalent Isotropically Radiated Power
Pt	=	Transmit Power
Gt	=	Antenna Gain
Lt	=	Loss in the Transmitting Equipment
A	=	Signal Attenuation
G/T	=	Antenna Figure-of-Merit
C/No	=	Signal Power to Noise Power Spectral Density
Eb/No	=	Signal Energy to Noise Power Density
Rb	=	Bit Rate
QPSK	=	Quadrature Phase Shift Keying
RS	=	Reed-Solomon Code

I. Introduction

A number of next generation satellite communication systems that utilize higher frequency bands, such as the Ka-band, and support spot-beam technology, on-board packet processing, and on-board packet switching are currently under development. These new systems will allow higher data rates than today's satellites, and will enable the use of smaller, low-power, and low-cost user terminals. This will allow satellite communication systems to become more competitive against other broadband communication solutions (cable, digital subscriber line) in providing integrated voice, data, and multimedia communications [1].

[†] Staff Scientist, Patton Electronics, Gaithersburg, MD 20879, USA (e-mail: gakkor@patton.com). The author was with the Department of Electrical and Computer Engineering, University of Maryland at the time of this work.
[‡] Professor, Department of ECE and ISR, University of Maryland, College Park, (e-mail: baras@isr.umd.edu).
[•] Assistant Research Scientist, ISR, University of Maryland, College Park (e-mail: michalis@isr.umd.edu).

The wide-area coverage, direct and ubiquitous access capabilities put these systems in an advantageous position for supporting applications that require concurrent transmission of the same content to multiple users. One such application is the reliable delivery of large data to multiple user locations, such as the distributed software, database and inventory updates. In order to achieve reliable delivery, missing or corrupted data segments must be detected and recovered. Traditionally, this has been achieved by automatic-repeat-request (ARQ) based schemes, where users detect missing or corrupted segments, and automatically request for retransmissions by returning feedback messages to the sender. However, it has been shown in the literature that for multicast content delivery over satellite networks, pure-ARQ based schemes do not perform favorably for several reasons [2]. First of all, collecting feedback messages from a large number of users results in the well-known *feedback implosion* problem. Feedback implosion is identified as a major problem when a large number of users try to transmit their feedback messages to a central node, causing high traffic concentration and backlog in the network. Moreover, long propagation delays (a round-trip-time (RTT) of 500ms for a two-way geo-synchronous earth orbit (GEO) satellite system) and highly variable propagation conditions (deep fading due to rain and atmospheric attenuation at Ka-band frequencies) associated with the satellite channels make repeat-request based recovery mechanisms very inefficient. Lastly, retransmission of individual data segments in response to individual user requests does not make good use of the broadcast capability of the satellite channel, because the retransmitted segment only benefits a few users.

Although it is not possible to abandon an ARQ-based scheme completely, shortcomings of these schemes have led to alternative designs that act proactively to protect transmitted data against the losses, and therefore avoid costly retransmissions over the satellite link. Packet level forward error correction (FEC) coding has been widely accepted as an efficient way of reducing the number of retransmissions and recovering from packet losses [3, 4]. While there has been a significant effort in defining packet level FEC schemes that detail the structure, organization and delivery of the encoding packets, little attention has been given to how the physical channel, channel coding, and the downlink power budget affect the design space of these schemes in the context of satellite network architecture.

We believe that in order to gain a better understanding of the potential benefits of packet level FEC coding in reliable multicast content delivery over satellite networks, the cross-layer interaction should be incorporated into the design and analysis of these schemes. Therefore, in this paper, we take a step in this direction, and define the points of interaction between various components and build a simulation environment that incorporates a model for the Ka-band channel, a detailed link-budget calculation for the satellite downlink, and a model for the underlying channel encoding for a Digital Video Broadcast (DVB) based packet delivery scheme. Using this simulation setup, we provide design suggestions for integrating FEC coding into reliable multicast transport protocols.

The rest of the paper is organized as follows. In Section II, we describe our network architecture. In Section III, we describe the network stack and define how each layer of stack interacts with the other layers. We also introduce the corresponding models for each layer that will be used in the remainder of this paper for analysis. Section IV discusses the design issues related to the integration of packet level FEC coding to a generic reliable transport protocol. Last section concludes the paper.

II. Network Architecture

We assume that the multicast content (software, file) is located at a server on an IP based terrestrial network. This content is to be distributed reliably to multiple user locations, which are within the coverage area of the satellite network. In this scenario, users that are equipped with two-way direct communication terminals access the terrestrial network through a gateway node that we will refer to as the Network Operations Center (NOC). Therefore, the NOC is responsible for fetching the content from the terrestrial server, and then distributing it over the satellite network (Fig. 1).

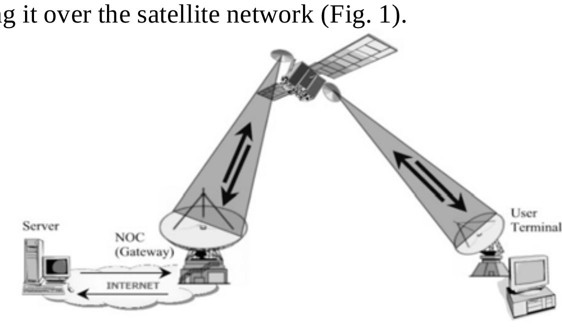

Figure 1. Satellite network architecture. Multicast content is distributed to multiple user locations through the NOC.

We assume that the session is established using the idea of connection splitting [5] that has recently been of much interest to the research community. In this scenario, the session is split into three components: (a) a TCP/IP connection is established between the server and the satellite gateway, (b) over the satellite portion of the network, a satellite reliable multicast transport protocol is run, and (c) between the user terminal and the user machines another TCP/IP connection is established. We assume that data reliability is ensured natively by the TCP protocol in sub-connections (a) and (c). Therefore, we are primarily interested in the reliable delivery of multicast content over the satellite portion of the network (Fig. 2). Our decision to go with the idea of connection splitting in this type of a network scenario is motivated by several observations. The first one stems from the particular structure of the network topology where the connections from the server to the gateway, and from the user terminal to the user machines are essentially point-to-point. Inside the satellite core, on the other hand, the star topology allows satellite to push content in single hop to all users. Secondly, the transmission characteristics, such as the transmission delay and error patterns, over the satellite portion are widely different than those that are over the terrestrial network. Finally, satellite service providers are more likely to favor this setup than to an end-to-end solution between the server and user machines, since the former allows them to have total control over the traffic flow inside their network for activities ranging from resource allocation, and security to billing.

Under this setup, we assume that Digital Video Broadcasting (DVB) [6] standard for data broadcasting is implemented for delivering data packets inside the satellite core network, although the discussion in the remainder of this paper does not depend on the behavior of a particular packet delivery scheme. DVB standard for data broadcasting allows encapsulation of IP datagrams into MPEG-2 Transport Stream

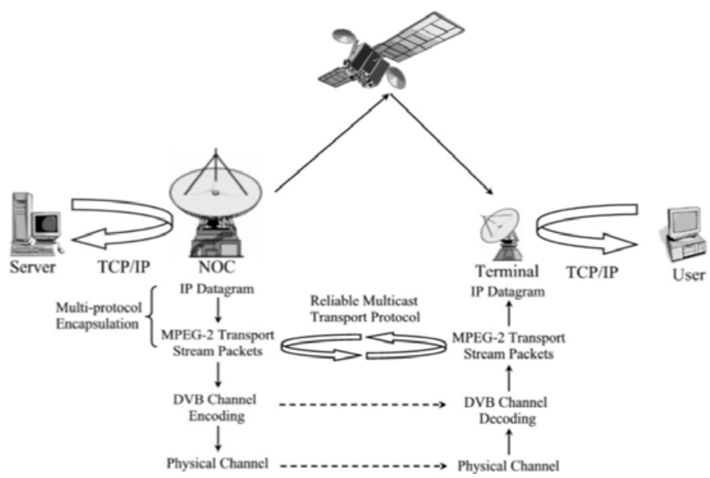

Figure 2. The illustration of the connection splitting idea and the overview of the network stack at the satellite core network.

Packets using the Multiple Protocol Encapsulation (MPE) standard [6]. The incoming stream of IP datagrams to the NOC are segmented into 188-Bytes (4 Bytes packet header and 184 Bytes payload) transport packets and buffered. A generic reliable multicast transport protocol delivers the transport stream packets and ensures the reliability over the DVB stack (Fig. 2). This network architecture will act as a template for the rest of the discussion in this paper.

In this paper, our goal is not to define a new multicast transport protocol or a FEC coding scheme, but rather to investigate the interaction between the different levels of the network stack and to provide design suggestions to facilitate the integration of FEC schemes into a given satellite architecture. To this extend, the next section describes the important components of the network stack and the corresponding models that we are going to use for our analysis.

III. Network Stack

The success of packet delivery at the transport layer depends heavily on the behavior of several underlying layers and the cross-layer interaction between them. The characteristic of the physical channel, along with the downlink power budget of the satellite system directly affect the number and the pattern of packet losses experienced by the transport protocol. The channel encoding and decoding determine the extent of packet losses that the system can recover from before the packet recovery mechanisms of the transport protocol. In the following subsections we describe these components and the corresponding

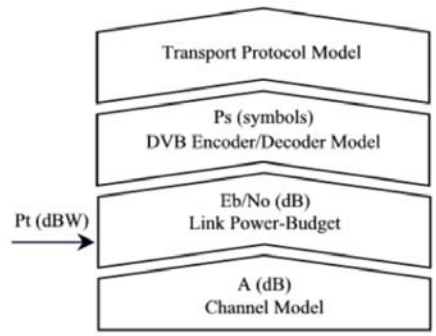

Figure 3. An abstract view of the network stack and an illustration of how different layers are coupled to each other.

models for them (Fig. 3).

A. Channel Model

Next generation systems are moving in the direction of using higher frequency bands, because higher bands offer wider bandwidth segments that are not available at more crowded lower frequency bands. Therefore, we consider a Ka-band channel in our evaluations. The Ka-band frequencies are very susceptible to attenuation due to rain and atmospheric effects. Therefore, the variation of the channel attenuation level is the first parameter of interest in coupling various components of the network stack.

In order to determine the characteristic of the rain attenuation for the Ka-band channel, we use a model that is based on the simulator developed at DLR (German Aerospace Center), Institute for Communications and Navigation [7]. The model is based on specific channel model parameters from the DLR measurement campaign carried out at Oberpfaffenhofen near Munich, Germany, in the years 1994 till 1997 with the 40 GHz beacon of the Italian satellite ITALSAT. The channel simulator generates a time-series of rain attenuation, and calculates the cumulative distribution of attenuation. It is also possible to extract the probability of being in a fade exceeding a given duration and exceeding a fading depth given as parameter (Fig. 4). The time series generator of the simulator outputs an attenuation level sample with a time resolution of 64 seconds. The attenuation level, A in dB, is the parameter that links the channel model to the link power-budget calculation of the satellite downlink. The following section presents the link-budget calculation for the system.

B. Link Power-Budget

The downlink power-budget determines the power margin that the satellite system has against signal attenuation. For a given channel attenuation level, the link budget gives the signal-to-noise ratio figure at the receiver terminal. This signal-to-noise ratio figure is the link that relates the link budget to the performance of the DVB channel encoding and decoding scheme.

In this paper, the link power-budget calculation is adapted from an application for a commercial satellite system [8]. For a given transmit power P_t in decibel Watts (dBW), the Equivalent Isotropically Radiated Power (EIRP) for the antenna system in dBW is given by

$$EIRP = Pt + Gt - Lt, \quad (1)$$

where G_t and L_t are the antenna gain, and the losses in the transmitting equipment in decibels (dB), respectively. The losses due to signal propagation through the atmosphere and rain attenuation are calculated as

$$A = Ap + Ar, \quad (2)$$

where, A_p and A_r are the losses due to propagation, and rain attenuation, respectively, both in dB. Then, the ratio of signal power to noise power spectral density in decibel Hertz (dBHz) follows as

$$C/No = EIRP - A + G/T - k, \quad (3)$$

where G/T in decibels per Kelvin (dB/K) is called the figure-of-merit of the receiver determined by the antenna gain G (dB) and its overall noise temperature T in Kelvin (K), and k is the Boltzmann constant in dBW/K/Hz. For a bit rate of Rb in dBHz, the ratio of bit energy to noise power density becomes

$$Eb/No = C/No - Rb \text{ in dB.} \quad (4)$$

The attenuation due to rain and atmospheric effects becomes substantial at Ka-band frequencies, and is the most important factor. Therefore, we assume that all other effects remaining constant we can express Eb/No as a function of the transmit power P_t and the attenuation level A for a given downlink rate. Consequently, one can rewrite Eq. (4) as follows:

$$Eb/No = Pt - A + \beta, \quad (5)$$

where $\beta = Gt - Lt + G/T - k - Rb$. This bit energy to noise power density ratio can be used to determine the bit error rate (BER) of the physical channel at the input of the DVB channel decoder for a given signal modulation scheme. In the remainder of this paper, we assume that the component of channel attenuation due to signal propagation (Ap) remains constant, while the rain-fading channel simulator described in the previous section determines the rain attenuation (Ar) component. The performance of the DVB channel coding is discussed in the next section. The link budget parameter values to calculate β are given in Table 1.

Gt (dB)	**Lt (dB)**	**Ap (dB)**
46.50	0.50	210.75
G/T (dB/K)	**k (dB/K/Hz)**	**Rb (dB Hz)**
16.37	-228.60	79.64

Table 1. Numerical values of the link-budget parameters used for the calculation of the downlink power budget.

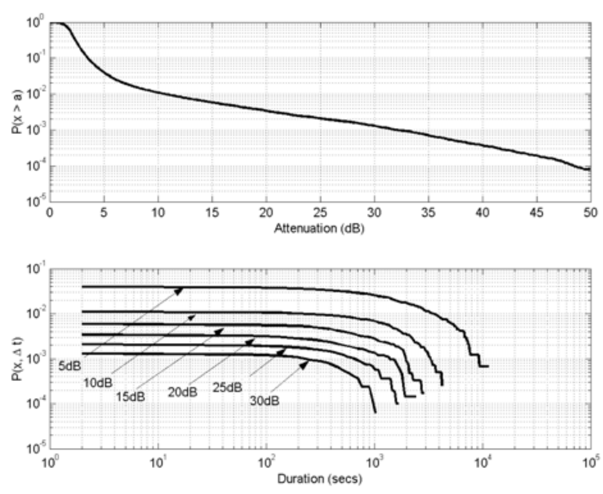

Figure 4. (a) Probability of exceeding a given channel attenuation level. (b) Probability of having a given fade durations at a given fade level.

C. DVB Channel Encoding and Decoding

On the transmitter side, the DVB channel encoder encodes each 188-Byte MPEG-2 transport stream packet, first using a shortened (204,188) RS code (adds 16 parity Bytes to each transport packet), and then, passing the packet bit stream through an interleaver. The interleaved symbols are encoded by a rate punctured convolutional encoder capable of supporting variable coding rates of 1/2, 2/3, 3/4, 5/6, and 7/8, and transmitted through the physical channel. On the receiver side, the bit stream is first decoded using a Viterbi decoder, and then passed through a de-interleaver. The de-interleaved symbols are then decoded using a RS decoder (Fig. 5).

In this section, we describe the analytical performance of the DVB channel decoding. The following analysis has been widely used in the literature and has been shown to approximate the simulation results closely [9]. The convolutional code performance is approximated by the union bound, which is truncated after a significant number of terms:

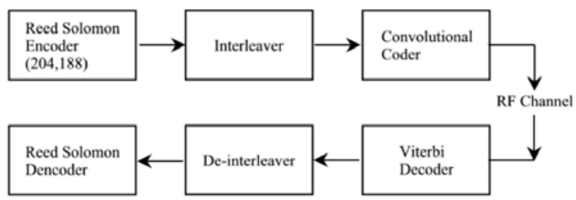

Figure 5. Abstract view of the channel encoding and decoding scheme for concatenated coding.

$$P_c \approx \frac{1}{k} \cdot \sum_{d=d_{free}}^{d_{free}+N} c_d \cdot P_d \,, \tag{6}$$

where k is the number of bits input to the convolutional encoder, d_{free} is the free distance of the code, c_d is the number decoder (trellis) paths that differ from the correct path at exactly d positions, P_d is the probability of incorrectly choosing such a path, and N is the number of terms used in the calculation of the union bound. The values for d_{free} and c_d are tabulated in Table 2. For coherent QPSK modulation and demodulation, P_d is given by:

$$P_d = \frac{1}{2} \cdot \text{erfc}\left(\sqrt{d \cdot R \cdot \frac{E_b}{N_o}}\right), \tag{7}$$

where, R is the overall rate of the concatenated code, Eb/No is the bit energy to noise power density ratio, and erfc(.) is the complementary error function. Then, assuming that the interleaver successfully breaks apart the long error burst at the output of the Viterbi decoder, the probability of symbol errors at the output of the Reed-Solomon decoder can be upper-bounded by:

$$P_s \leq b \cdot P_c \,, \tag{8}$$

where, b is the symbol length of the RS code in bits. This probability of symbol error value is the parameter that links the lower levels to the transport protocol level since corrupted packets have to be recovered at this level.

Code	**½**	**2/3**	**¾**	**5/6**	**7/8**
$\mathbf{d_{free}}$	10	6	5	4	3
$\mathbf{c_d}$	[36, 0, 211, 0, 1404, 0, 11633, 0, 77433, 0, 502690, 0, 3322763, 0, 21292910, 0, 134365911, 0]	[3, 70, 285, 1276, 6160, 27128, 117019]	[42, 201, 1492, 10469, 62935, 379644]	[92, 528, 8694, 79453, 792114]	[9, 500, 7437, 105707, 1402743]

Table 2. The free distance and cd values for the rate punctured convolutional codes used by the DVB encoder.

IV. Design Issues for Packet Level FEC Coding Integration

Having defined our network architecture, elements of the network stack, and the corresponding models for them, in this section, we discuss the design issues related to the integration of FEC coding. In general, a

FEC encoder generates n encoding packets from k original data packets, and the original k data packets can be recovered as long as $k \cdot (1+\varepsilon)$ out of the transmitted n encoding packets are received correctly, for some $\varepsilon \geq 0$, representing the decoding overhead. One immediate benefit of this construction is that it allows the recovery of the original data packets even if some of the encoding packets are lost. Therefore, the number of packet retransmissions is reduced. However, it also reduces the bandwidth efficiency of the system. Consequently, one important design question deals with how the transmission of the encoding packets is organized. Another benefit of packet level FEC coding is that a transmitted encoding packet benefits all receivers equally in accumulating enough packets for decoding. Therefore, individual packet retransmission to individual receivers is avoided and system makes good use of the broadcast capabilities of the satellite channel. However, if the system runs out of encoding packets before all receivers accumulate enough packets to start decoding, then the system would have to resort to retransmissions to complete the transmission. Hence, the number, n, of encoding packets that can be generated for a given set of input packets is another important design criterion. Similarly, number of input packets, k, is also an important design factor since it effects the decoding and processing delays encountered by the system.

We believe that these and similar questions can be best answered if the transport protocol design takes a bottom-to-top approach and considers the cross-layer interaction between the different network layers. To this extend, the first observation we make involves the effects of having a downlink power margin and adaptive code rates for the convolutional coding.

DVB Channel Encoding	Required Eb/No (dB)	Rain Fade Margin (dB)
RS & CC Rate 1/2	3.62	9.97
RS & CC Rate 2/3	4.00	9.58
RS & CC Rate 3/4	4.50	9.09
RS & CC Rate 5/6	5.00	8.59
RS & CC Rate 7/8	5.37	8.20

Table 3. Sustainable rain fade margins to satisfy a BER of 10^{-9} (given in terms of required Eb/No values) at the output of the DVB channel decoder for a downlink transmit power of Pt = 20W.

In Table 3, we list the sustainable rain fade margins to satisfy the BER target of our system for a given transmit power of Pt = 20W, and a downlink rate of Rb = 92 Mbps (79.64 dB Hz). We observe that, by changing the code strength, it is possible to sustain the target system performance for attenuation levels of

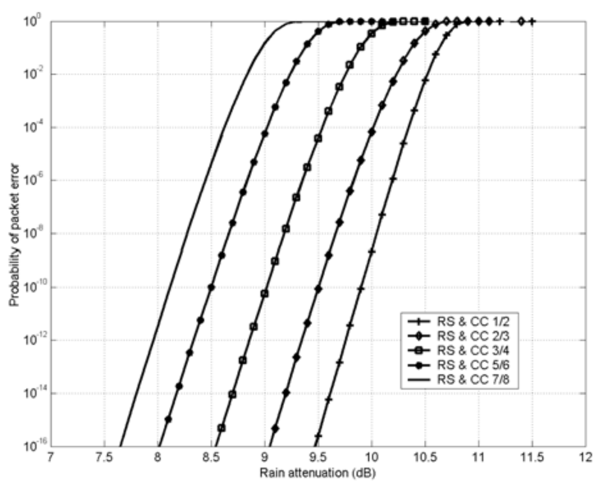

Figure 6. Sustainable attenuation levels for different channel encoding strengths.

up to 10dB. However, beyond this value, the code performance deteriorates rapidly and the probability of bit error goes to one (Fig. 6).

Linking this observation to the statistical behavior of the underlying channel, we observe from Fig. 4(a) that probability of exceeding an attenuation level of 8dB is approximately, P(x>8dB) = 0.0153. Therefore, most of the time the system would be operating at favorable conditions where the bit error rate is kept at very low values by the help of channel coding. Although this threshold is rarely exceeded, from Fig. 4(b), we observe that there is a fairly uniform probability of seeing fade durations of 10-100 seconds. Considering that it takes approximately 35.5µsecs to transmit a 188-Byte MPEG-2 Transport Stream Packet coded using the (204,188) RS code and rate-½ convolutional code at a downlink rate of 92Mbps, a fade duration on the order of seconds would result in the loss of all or very large portion of the packets of a given session.

Following these observations, we run the following test. We assume that data to be distributed can be organized into B batches of k input packets, where each packet has size l = 188 Bytes. The distribution of the packets follows a round robin scheme, where packets are transmitted in an interleaved fashion, one from each batch until the first B·k packets are forwarded. This pattern distributes long error burst among packets of different batches, and allows completion of the original set of packets with minimum delay by the receivers that have favorable channel characteristics. Following the distribution of the initial set of packets, feedback (status reports) is collected from receivers on the success of the initial distribution. Since the channel state is expected to be favorable for most of the receivers, we expect that at this point only a few receivers will request additional packets.

In Table 4, we show the success profile of 100 users after the distribution of the initial set of packets for set sizes of 0.188, 1.88, and 18.8 MBytes. The system is assumed to be operating with a downlink transmission power of Pt=20W, and channel encoding of RS (204,188) + CC 7/8. The effective rate-share of the session at the transport layer is 2Mbps. The initial distribution terminates after all the receivers return their status reports. The test is repeated for 1000 distributions for each set size and average results are reported.

File Size (Batch, Batch Size)	**0.188MB (B=100, k=10)**	**1.880MB (B=100, k=100)**	**18.8MB (B=100, k=1000)**
Successful users (%)	99.766	99.762	99.721
Failed users with no packet accumulation (%)	84.352	81.738	57.662
Download time (secs)	1.2520	8.0200	75.7000

Table 4. Success profile of 100 users after the distribution of initial set of packets.

As expected after initial distribution, a very high percentage (99.7%) of all users complete the download successfully *without requesting* any additional packets. This is achieved using the *lowest strength* channel-encoding scheme. The session download time is given by a single round-trip-time propagation delay (500ms) in addition to the transmission delay. More importantly, among the users that failed to complete the download following this initial distribution, a high percentage of them fail to accumulate any packet. This reinforces our previous observation that the channel encoding and power margins are powerful enough to keep the channel error free for most of the time, and in the remaining cases the fading is too deep to recover any packets. These leads to following two properties:

Property 1: The FEC coding, if any, should be systematic, meaning that the first encoding packets generated should be equal to the source packets. The probability that all receivers experience attenuation levels that are higher than the threshold described above is very small. Therefore, most receivers would complete reception of packets without requiring additional packets. Consequently, having a systematic code would avoid unnecessary decoding and buffering delay.

Property 2: The scheme for proactive protection of transmitted packets has to be adaptive to the changing channel conditions. Having fixed protection would be a waste of resources since the system will be operating at favorable conditions for most of the time.

Following this initial attempt, we assume that the protocol starts transmitting additional encoding packets for each batch in order to assist in the recovery process. The recovery process progresses in rounds, where in each round the protocol transmits additional number of encoding packets for each batch based on the maximum requested to complete the batch by the users. Since the additional encoding packets can also be lost due to corruption, at the end of each recovery round, the protocol collects feedback from the users and continues to transmit additional encoding packets until all users complete the download. In order to determine the requirements on the encoding under these assumptions, we run the following test. For each file size of 0.188, 1.88, and 18.8 Mbytes, we look at 1000 different downloads by 100 users. The system is assumed to be operating with a downlink transmission power of Pt=20W, and channel encoding of RS (204,188) + CC 1/2. The effective rate-share of the session at the transport layer is 2Mbps. We record the number of times that more than one attempt is required, the average number of attempts, the success ratio after at most 50 attempts, and the corresponding download times. These results are tabulated in Table 5.

File Size (Batch, Batch Size)	Number of downloads with > 1 attempts (%)	Average (max) number of additional attempts	Average (max) time (seconds)	Average number of successful users (%)
0.188MB (B=100, k=10)	14.89	4.3875 (50)	4.1547 (56.5539)	99.90
1.880MB (B=100, k=100)	14.11	5.5978 (50)	26.3627 (396.2627)	99.89
18.8MB (B=100, k=1000)	16.54	13.7603 (50)	382.5299 (3784.3833)	99.93

Table 5. User profile after additional attempts for recovery of the corrupted packets.

We observe that, of the total of 1000 downloads, only 14-16% of them require the use of additional rounds by the users. However, to complete these instances, the protocol has to make several rounds and in some instances cannot complete the download after the maximum of 50 additional attempts is reached, even though the strongest strength channel coding is used. This is also evident from the increased session time. We conclude that the use of additional encoding packets helps some of the users to complete their downloads, but still cannot guarantee 100% success in all cases because the users remain in deep fade for several minutes. These observations lead to following property:

Property 3: The scheme should be able to withstand bursty packet losses, the duration of which may be on the same order of that of the transmission session. Therefore, the number, n, of encoding packets that can be generated for each block of input packets has to be much larger the number of source packets, k. Additionally, the scheme should be able to create encoding packets on-demand, rather than a fixed number, in order to cope with rare attenuation events which have long durations.

Although there exist coding schemes that satisfy Property 3 up to certain extend [10], our observations indicate that the protocol would be better of by terminating the download after a few additional attempts, and repeating the download at a later time for the remaining users (that constitute a very small percentage of the original group). In the next section, we summarize our findings.

V. Conclusions

In this paper we introduced a simulation environment to investigate the cross-layer interaction between different components of the network stack, and their effect on the design parameters of FEC coding for reliable data multicast. We conclude that, strong channel encoding and power margins in the system design keep system performance at favorable levels for most of the operational time. This makes lower level errors transparent to the transport protocol. Therefore, such a system would not require fixed protection, or highly proactive mechanisms that transmit additional encoding packets in advance. However, the coding should be able to generate additional encoding packets on-demand to recover losses when the system is in transition from a favorable channel environment to a high error rate channel environment. Finally, when some users are in deep fade periods, we conclude that full recovery is not possible with in reasonable session times even if a FEC code capable of generating many encoding packets is used. For such instances, it is better for the transport protocol to drop the connection to such users rather than wasting system resources with multiple attempts.

Acknowledgments

This material is based upon work supported by NASA under award number NCC8235. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.

References

1. J. V. Evans, "Communication satellite systems for high speed Internet access," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 5, October 2001.

2. G. Akkor, M. Hadjitheodosiou, and J. S. Baras, "Transport protocols in multicast via satellite," *Int. J. Sat. Comm.*, vol. 22, no. 6, pp. 611-627, November\December 2004.

3. M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, "Forward error correction building blocks," *Internet Society Request for Comments*, December 2002, RFC3452.

4. M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, "The use of forward error correction in reliable multicast," *Internet Society Request for Comments*, December 2002, RFC3453.

5. X. Zhou, X. Liu, and J. S. Baras, "TCP over GEO satellite hybrid networks," in *Proc. of IEEE MILCOM*, vol.1, pp. 29-34, October 2002.

6. "Digital Video Broadcasting (DVB): DVB specification for data broadcasting," *European Telecommunication Standards Institute*, ETSI EN 301 192 v.1.3.1, 2003.

7. U.-C. Fiebig, "A time-series generator modeling rain fading and its seasonal and diurnal variations," in *Proc. of 1st International Workshop of COST-Action 280*, Malvern, UK, 2002.

8. E. J. Fitzpatrick, "SPACEWAY system summary," *Space Communications*, vol. 13, pp. 7-23, 1995.

9. J. Foerster, and J. Liebetreu, "FEC performance of concatenated Reed-Solomon and convolutional coding with interleaving," *IEEE 802.16 Broadband Wireless Access Working Group* http://ieee802.org/16, 2000.

10. M. Luby, "LT codes," in *Proc. of IEEE Symposium on Foundations of Computer Science*, pp. 271-280, November 2002.